

# 8장. 제어문

## 8장. 제어

---

### 8.1 블록문

1. 블록문의 끝에는 세미콜론을 붙이지 않는다.

### 8.2 조건문

#### 8.2.1 if ... else 문

1. if문 기본 형식

```
if (조건식1){  
  
} else if(조건식2){  
  
} else {  
  
}
```

#### 8.2.2 switch 문

1. if ... else 문의 조건식은 Boolean 값으로 평가되어야 하지만, switch 문은 주로 String, Number 사용

2. [ 예제 08 - 07 ]

```
var month = 11;  
var monthName;
```

```

switch (month) {
  case 1: monthName = "January";
  case 2: monthName = "February";
  case 3: monthName = "March";
  case 4: monthName = "April";
  case 5: monthName = "May";
  case 6: monthName = "June";
  case 7: monthName = "July";
  case 8: monthName = "August";
  case 9: monthName = "September";
  case 10: monthName = "October";
  case 11: monthName = "November";
  case 12: monthName = "December";
  default: monthName = "Invalid month";
}

console.log(monthName); // Invalid month

```

- 해결방법 : case의 해당하는 문 마지막에 **break**를 추가 ( default 문에서는 생략 )

## 8.3 반복문

### 8.3.1 for 문

#### 1. [ 예제 08 - 13 ]

```

for ( var i = 0; i <= 6; i++ ){
  for ( var j=0; j <= 6; j++ ){
    if ( i+j === 6 ) console.log(`[${i}, ${j}]`);
  }
}

```

### 8.3.2 while 문

1. while 문은 조건문의 평가 결과가 거짓이 되면. 코드 블록을 실행하지 않고 종료한다.
  - 만약 조건식이 Boolean 값이 아니면, Boolean 값으로 강제 변환하여 참, 거짓을 구별한다.

## 2. 무한루프

```
while (true) { ... } // 무한루프
```

### 8.3.3 do ... while 문

1. 코드 블록을 먼저 실행하고, 조건식을 평가한다.

➡ 코드 블록은 무조건 한 번 이상 실행됨,

```
var count = 0;

// count가 3보다 작을 때까지 코드 블록을 계속 반복 실행
do {
  console.log(count); // 0 1 2
  count++;
} while (count<3);
```

## 8.4 break 문

1. 레이블 문(label statement) : 식별자가 붙은 문

2. [ 예제 08 - 19 ]

```
foo : console.log('foo'); // foo라는 레이블 식별자가 붙은 레이블
```

3. 중첩된 for 문 외부로 탈출할 때 유용, 그 밖의 경우에는 일반적으로 권장 X

➡ 레이블 문을 사용하면, 프로그램 흐름이 복잡해져서 가독성↓, 오류 발생 가능성↑

## 8.5 continue 문

1. 반복문의 코드 블록 실행을 현 지점에서 중단하고 반복문의 증감식으로 실행 흐름을 이동시킴.